

IN THE CLAIMS:

- 1 1. (ORIGINAL) A method for a network node, which includes a central processing unit
- 2 (CPU) configured to execute a router operating system, to filter malicious data packets
- 3 received at the network node, the method comprising:
 - 4 receiving a data packet at the network node;
 - 5 performing hash-based flow classification on the received data packet to deter-
 - 6 mine whether the received data packet is a malicious data packet; and
 - 7 discarding the received data packet before the data packet can be forwarded to the
 - 8 CPU for processing by the router operating system, if the received data packet is deter-
 - 9 mined to be a malicious data packet.
- 1 2. (ORIGINAL) The method of claim 1, wherein the step of performing hash-based flow
- 2 classification further comprises:
 - 3 identifying a packet type associated with the received data packet;
 - 4 extracting a set of signature information corresponding to the identified packet
 - 5 type; and
 - 6 searching a hash table to locate the extracted set of signature information.
- 1 3. (ORIGINAL) The method of claim 2, further comprising:
 - 2 configuring the hash table, either manually or automatically, to associate the set of
 - 3 signature information with a data flow; and
 - 4 determining whether the data flow associated with the set of signature information
 - 5 corresponds to a malicious data flow.
- 1 4. (ORIGINAL) The method of claim 1, further comprising:
 - 2 associating the received data packet with a destination in the network node as a
 - 3 result of the hash-based flow classification.

1 5. (ORIGINAL) The method of claim 4, further comprising:
2 determining whether the destination associated with the received data packet is a
3 predetermined destination associated with malicious data packets.

1 6. (ORIGINAL) The method of claim 5, further comprising:
2 in response to determining that the destination associated with the received data
3 packet is the predetermined destination, performing the steps of:
4 removing buffer pointers from a set of descriptors associated with the re-
5 ceived data packet; and
6 storing the removed buffer pointers on a queue of free buffer pointers.

1 7. (ORIGINAL) The method of claim 6, further comprising:
2 if the queue of free buffer pointers does not contain enough available entries to
3 store the removed buffer pointers, storing the set of descriptors associated with the re-
4 ceived data packet on a delete queue until enough entries become available in the queue
5 of free buffer pointers.

1 8. (ORIGINAL) The method of claim 6, further comprising:
2 transferring free buffer pointers from the router operating system to the queue of
3 free buffer pointers.

1 9. (ORIGINAL) The method of claim 1, wherein the step of performing hash-based flow
2 classification is used in conjunction with an access control list or an intrusion detection
3 system.

1 10. (ORIGINAL) The method of claim 1, wherein the network node is an intermediate
2 network node.

1 11. (ORIGINAL) A network node, comprising:

2 a central processing unit (CPU) configured to execute instructions that implement
3 a router operating system;
4 a network interface adapted to receive a data packet;
5 a memory having a plurality of storage locations addressable by the CPU, the
6 storage locations being configured to store:
7 (i) at least a portion of the router operating system instructions,
8 (ii) one or more data buffers for storing the received data packet, and
9 (iii) a searchable data structure configured to store information associ-
10 ated with the received data packet; and
11 a system controller coupled to the memory and the CPU, the system controller
12 including a hardware assist (HWA) module configured to discard malicious data packets
13 from the network node before the malicious data packets can be forwarded to the CPU
14 for processing by the router operating system.

1 12. (ORIGINAL) The network node of claim 11, wherein the searchable data structure is
2 a hash table.

1 13. (ORIGINAL) The network node of claim 11, wherein the HWA module includes a
2 direct memory access (DMA) controller and a flow classifier.

1 14. (ORIGINAL) The network node of claim 13, wherein the DMA controller includes:
2 an ingress descriptor first in, first out (FIFO) queue configured to store a set of
3 descriptors referencing the one or more data buffers in which the received data packet is
4 stored;
5 a packet-header buffer configured to store information contained in at least one
6 packet header prepended to the received data packet;
7 an egress descriptor FIFO configured to store the set of descriptors as well as a
8 data flow identification (ID) value for identifying the data flow associated with the re-
9 ceived data packet and a destination value for identifying a destination in the network

10 node associated with the received data packet, the flow classifier searching the searchable
11 data structure to locate the data flow ID value and the destination value; and
12 a free-buffer FIFO containing a set of free buffer descriptors allocated for the
13 network interface.

1 15. (ORIGINAL) The network node of claim 13, wherein the flow classifier includes:
2 a packet-identifier engine configured to identify a packet type associated with the
3 received data packet based on information received from the DMA controller;
4 a signature-extraction engine configured to extract a set of signature information
5 from a predetermined set of fields in the information received from the DMA controller,
6 the predetermined set of fields being selected based on the packet type identified by the
7 packet-identifier engine;
8 an address generator configured to generate a memory address based on the set of
9 signature information, the memory address corresponding to an entry in the searchable
10 data structure; and
11 a search module configured to search the searchable data structure to locate a flow
12 ID value and a destination value associated with the received data packet.

1 16. (ORIGINAL) The network node of claim 15, wherein the flow classifier further in-
2 cludes:
3 an egress packet manager configured to reformat descriptors from an ingress de-
4 scriptor format to an egress descriptor format.

1 17. (ORIGINAL) The network node of claim 11, wherein the network node is an inter-
2 mediate network node.

1 18. (ORIGINAL) A network node including a central processing unit (CPU) configured
2 to execute a router operating system, the network node comprising:
3 means for receiving a data packet at the network node;

4 means for performing hash-based flow classification on the received data packet
5 to determine whether the received data packet is a malicious data packet; and
6 means for discarding the received data packet before the data packet can be for-
7 warded to the CPU for processing by the router operating system, if the received data
8 packet is determined to be a malicious data packet.

1 19. (ORIGINAL) A computer-readable media including instructions for execution by a
2 processor, the instructions for a method of filtering malicious data packets received at a
3 network node in which a central processing unit (CPU) is configured to execute a router
4 operating system, the method comprising:
5 receiving a data packet at the network node;
6 performing hash-based flow classification on the received data packet to deter-
7 mine whether the received data packet is a malicious data packet; and
8 discarding the received data packet before the data packet can be forwarded to the
9 CPU for processing by the router operating system, if the received data packet is deter-
10 mined to be a malicious data packet.

1 20. (NEW) A method comprising:
2 receiving a data packet;
3 storing the data packet in one or more data buffers referenced by a set of one or
4 more packet descriptors;
5 passing the set of packet descriptors to an ingress descriptor queue at a direct
6 memory access (DMA) controller;
7 performing, by a flow classifier coupled of the DMA controller, hash-based flow
8 classification on the data packet referenced by the set of packet descriptors to associate
9 the data packet with a particular data flow identification (ID) value;
10 passing the set of packet descriptors to an egress descriptor queue; and

11 determining based on the data flow ID value if the data packet is a malicious data
12 packet and, if so, freeing the set of packet descriptors to discard the data packet before the
13 data packet can be processed by a router operating system operating on a CPU coupled to
14 the DMA controller and, if not, allowing the data packet to be processed by the router
15 operating system operating on the CPU coupled to the DMA controller.

1

1 21. (NEW) The method of claim 20 wherein freeing the set of packet descriptors com-
2 prises:

3 removing buffer pointers from the set of packet descriptors and placing the buffer
4 pointers on a free-buffer queue.

1 22. (NEW) The method of claim 21, further comprising:

2 if the free-buffer queue has insufficient space to store additional removed buffer
3 pointers, storing the set of descriptors in a delete queue until space becomes available in
4 the free-buffer queue.

1 23. (NEW) The method of claim 20 wherein allowing the data packet to be processed
2 comprises:

3 sending an interrupt to the CPU to notify the CPU that the set of packet descrip-
4 tors are available in the egress queue.

1 24. (NEW) The method of claim 20, wherein performing hash-based flow classification
2 comprises:

3 identifying a packet type associated with the data packet;

4 extracting a set of signature information corresponding to the identified packet
5 type; and

6 searching a hash table to locate the extracted set of signature information and ac-
7 cessing one or more associated entries.

- 1 25. (NEW) The method of claim 20 further comprising:
- 2 reformatting the set of packet descriptors from an ingress descriptor format to an
- 3 egress descriptor format before passing the set of packet descriptors to the egress descrip-
- 4 tor queue.